CSE443 – Object Oriented Design and Analysis

Homework 2

Report

OĞUZHAN AGKUŞ
161044003

Notes

I cannot completed the last question. Because electricity outage has occured in my area. It last very long. So, I could not charge my battery. I tried to use its last energy to write report for other parts. But it woud not enough.

Part 1

Question 1: What happens if someone tries to clone a Singleton object using the clone() method inherited from Object? Does it lead to the creation of a second distinct Singleton object?.

Answer 1: When someone tries to clone a Singleton object, s/he will get "the method clone() from the type Object is not visible." error.

Because clone() method in Object class is protected. So s/he have to override clone() method in Singleton class. If overriding is already done by doing:

```
protected Object clone() throws CloneNotSupportedException {
   return super.clone();
}
```

it will throw CloneNotSupportedException.

Question 2: Cloning Singletons should not be allowed. How can you prevent the cloning of a Singleton object?

Answer 2: Of course, cloning should not be allowed. It easy to avoid by overriding clone() method like below:

```
protected Object clone() throws CloneNotSupportedException {
 return uniqueInstance;
}
```

Question 3: Let's assume the class Singleton is a subclass of class Parent, that fully implements the Cloneable interface.

How would you answer questions 1 and 2 in this case?

Answer 3.1: Parent's clone() method will work and it will create a new Singleton object. This should not be allowed for Singleton class.

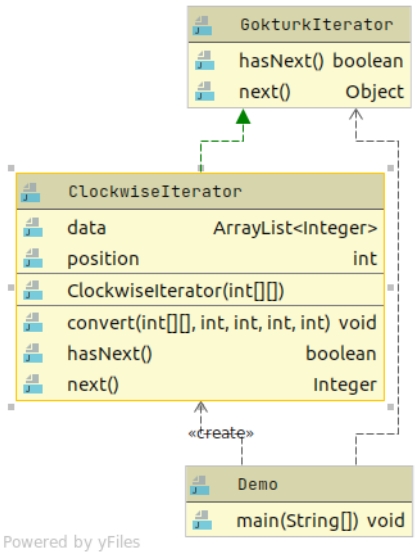
Answer 3.2: The clone() method must be override. It should return a reference to the Singleton object.

```
protected Object clone() throws CloneNotSupportedException {
   return uniqueInstance;
}
```

A test code snipped is attached, it is proof of these answers.

Part 2

In this part, I used iterator design pattern. The interface GokturkIterator is simple, it has next() and hasNext() methods only. My clockwise iterator implements it. Its constructor gets 2D matrix.



Class diagram

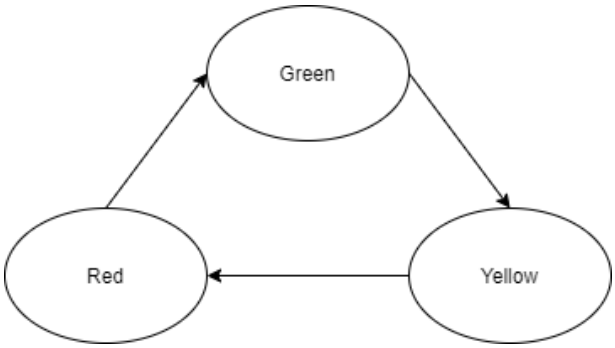
I test it with some examples. One of them:

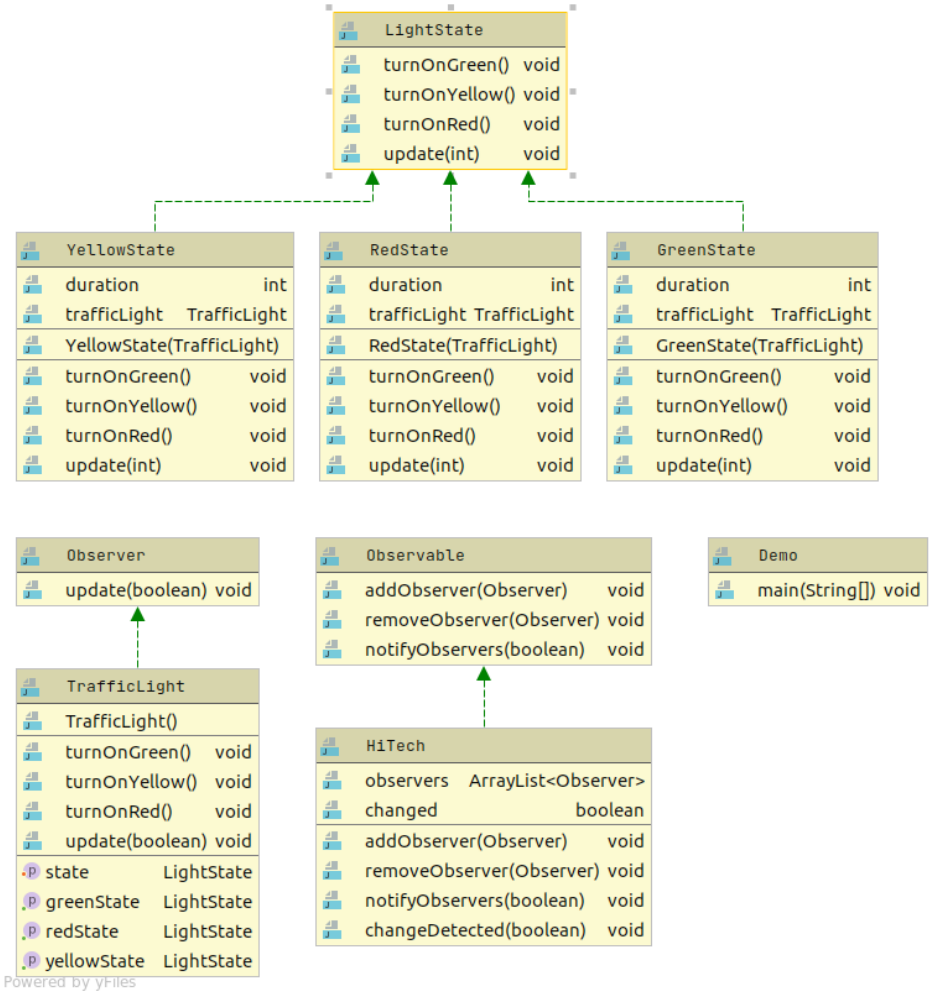
```
int[][] data = {{1,2,3,4,5},
                {6,7,8,9,10},
                {11,12,13,14,15}};
```

```
1 2 3 4 5 10 15 14 13 12 11 6 7 8 9
```

Part 3

In this part I used State and Observer patterns. The HiTech class represents Mobese cameras and it is observable object. The traffic light is observer obeject, it observe the HiTech object. The traffic light object has states for each color. The color change in this order.





Example output:

```
10:27:00.314154 - Green
10:28:00.332536 - Yellow
10:28:03.333258 - Red
10:28:18.335297 - Green
10:29:48.336107 - Yellow
10:29:51.337012 - Red
10:30:06.338293 - Green
10:31:06.339509 - Yellow
10:31:09.340698 - Red
```